ACTIVE MAINTENANCE OF HIERARCHICAL STRUCTURES IN FUTURE ARMY NETWORKS

Kyriakos Manousakis and Anthony J. McAuley
Telcordia Technologies
One Telcordia Drive, Piscataway, NJ, 08854, USA

ABSTRACT[1]

The application of appropriate hierarchical structures is an efficient mechanism for improving the survivability and scalability of mobile ad hoc networks (MANETs). Due to network dynamics and infrastructureless character of MANET, however, the imposed hierarchy may harm the performance as the network environment evolves. This degradation can be prevented by adapting the hierarchy to the new network characteristics, so that the resulting hierarchy continues to be beneficial. This hierarchy maintenance is traditionally localized and driven by the constraint violations (feasibility) due to network dynamics. The traditional class of hierarchy maintenance is called passive, since it is triggered only by infeasibility. In this paper we present a new class of hierarchy maintenance algorithms, called active hierarchy maintenance. Unlike the Passive approach, Active Maintenance continuously operate for both preserving the feasibility and improving the evolving hierarchical structures. Our active maintenance algorithms operate locally using the same basic objective functions used in hierarchy creation. For a specific class of objective functions, we prove that the localized reconfigurations applied by the active maintenance result in global hierarchy quality improvement. Using an example of creating balanced size domains, our simulation results show that Active Local Maintenance preserves good hierarchy quality (cost increase is less than 50% of the global optimal), independently of the network dynamics. To the contrary, Passive Local maintenance shows rapid quality degradation (cost increase up to 400%)

1. INTRODUCTION

Survivability and scalability of mobile ad hoc networks (MANETs) is important for the future army missions. On the other hand, the infrastructureless characteristics and dynamics of this class of networks results into the inefficient application of existing network protocols and complicate the problem of improving or inventing new protocols. For the effective application of MANETs, techniques and mechanisms have to be applied for improving protocol performance. One widely used mechanism is the organization of the network into smaller groups. The application of network protocols in hierarchical manner, provably results into more efficient handling of network dynamics in the imposed infrastructureless MANETs' environment. For example, hierarchy can reduce overall overhead (e.g., routing overhead with n nodes can be reduced from $O(n)$ to $O(n\log n)$). As important, the hierarchy can allow protocols to react much faster, better handle heterogeneity and simplify management.

The diversity of network environments and applications impose a variety of distinct performance requirements onto the hierarchy. The application of a specific hierarchical structure to be effective has to satisfy a set of imposed performance requirements resulting into the improvement of the corresponding set of network performance metrics. For example, some may emphasize reduced overhead (e.g., routing update messages) in order to optimize user capacity, others may give priority to minimizing routing stretch in order to minimize end-to-end delay. Obviously, the same hierarchical structures may be beneficial for one performance aspect yet harmful to some others.

The concept of different networks having diverse performance requirements or the same networks adapting to different set of objectives through their lifetime, have not been highlighted by the existing studies on hierarchy formation algorithms. There are many approaches to creating hierarchies, from traditional planning systems to automated clustering techniques in routing protocols. In ad hoc networks, for example, there are many distributed heuristics that have been used to create hierarchies. The most popular are based on creating Dominating Sets (Lin and Gerla, 1997; Basagni, 1999; Chatterjee et. al., 2002, Ramalingam et. al., 2002; Bao and Garcia-Luna-Aceves, 2003). These heuristics, however, are able to capture only a small subset of the hierarchy formation requirements, resulting in reduced hierarchy performance benefits. More flexible and effective are hierarchy creation tools that use general optimization techniques, such as Kernighan-Lin graph partitioning (Krishnan et. al., 1999) or Simulated

[1] Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance (CTA) Program, Cooperative Agreement DAAD19-2-01-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 NOV 2006**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Active Maintenance Of Hierarchical Structures In Future Army Networks**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Telcordia Technologies One Telcordia Drive, Piscataway, NJ, 08854, USA**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002075., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Annealing (Manousakis et. al., 2004) heuristics. However, due to their centralized nature these heuristics typically require more overhead and take longer to perform.

For establishing hierarchical structures with respect to the required performance objectives, the hierarchy formation algorithms have to be aware of these objectives. The quality quantification of the obtained hierarchical structures with respect to the imposed performance objectives is accomplished with the application of objective functions. These objective functions translate the quality of the obtained hierarchical map into a quantifiable value (cost). This value is utilized by the hierarchy formation algorithm to obtain the most appropriate structure.

In a MANET we need not only automatically create good hierarchies, but also maintain them as the network topology changes. To date, this hierarchy maintenance has been focused on the critical task of ensuring that the hierarchy does not violate any constraints (e.g. when a node is more than one hop from a cluster head (Basagni, 1999)). This passive event-driven approach minimizes reconfiguration and overhead, but results in an inevitable decline in network optimality, particularly in MANET environments. As it is shown in (Chatterjee et. al., 2002), the passive maintenance algorithms are not able to handle the performance requirements and network dynamics, resulting into inefficient hierarchical structures. To the contrary, this paper investigates a new active approach to maintaining a hierarchy, that can balance reconfiguration and signaling overhead with maintaining network optimality. The proposed active maintenance algorithms are aware of and adaptable to the imposed performance requirements. By utilizing localized information and the objective functions that represent the corresponding objectives, they continuously evaluate the quality of the localized hierarchy and attempt to improve it. For a specific class of objective functions we prove the localized decisions result into globally efficient hierarchical structures

Section 2 overviews an indicative set of objective functions we have developed for hierarchy quality quantification (Manousakis et. al., 2004). For maintainance, Section 3 describes the general mechanism of passive approaches (Lin and Gerla, 1997) and Section 4 describes our proposed active approach. Section 5 provides performance results that indicate the effectiveness of the approaches on satisfying the hierarchy formation requirements.

2. QUANTIFICATION OF HIERARCHY QUALITY

The quality quantification of the generated and maintained hierarchy is realizes in the form of objective functions. We have introduced (Manousakis et. al., 2004; Manousakis and Baras, 2004) objective functions for two classes of hierarchy formation requirements. The first class of requirements is related to the structural characteristics of the generated domains. For example:

Table I. Static Herarchy Objectives

Objective	Objective Function
Balanced size domains	$J(C)=\min_{C}\sum_{i-1}^{K}\lvert C_i\rvert^2$ (1)
Equal diameter domains	$J(C)=\min_{C}\sum_{i-1}^{K}\left(d_{C_i}^2\right)$ (2)
Distinct size requirement for each domain	$J(C)=\min_{C}\sum_{i-1}^{K}\left(\lvert C_i\rvert-\lvert C_i\rvert^*\right)^2$ (3)
Balanced size domains having also min BRs	$J(C)=\min_{C}\left[\sum_{i=1}^{K}\lvert C_i\rvert^2+10*\sum_{i=1}^{K}BR_{C_i}\right]$.(4)

The second class of requirements involves the mobility characteristics of the participating nodes. For example:

Table II. Dynamic Herarchy Objectives

Objective	Objective Function
Domain members move in a similar direction for robust domains	$J\left(C\right)=\min_{C}\sum_{z=1}^{K}\left(\sum_{i,j=1}^{\lvert C_z\rvert}\theta_{r_{i,j}}\right)^2$ (5)
Domain members have similar velocity for robust domains	$J\left(C\right)=\min_{C}\left(\sum_{z=1}^{K}\left[\sum_{i,j=1}^{\lvert C_z\rvert}U_{r_{i,j}^z}^2\right]^2\right)$ (6)
Robust links among domain members for robust domains	$J\left(C\right)=\min_{C}\left[-\sum_{z=1}^{K}\left(\sum_{i,j=1}^{\lvert C_z\rvert}\left(LET_{ij}\right)\right)^2\right]$ (7)

Where the definitions of the parameters that constitute the objective functions are provided below:

Parameter	Definition
C_i	Cluster *i*
$\lvert C_i\rvert$	Size of cluster *i*
$\lvert C_i\rvert^*$	Required size for cluster *i*
d_{C_i}	Diameter of cluster *i*
BR_{C_i}	Number of border routers of cluster *i*
$\theta_{r_{i,j}}$	Relative direction of nodes *i,j*
$U_{r_{i,j}}$	Relative Velocity of nodes *i,j*
LET_{ij}	Expiration Time of Link between nodes *i,j*

The corresponding objective functions (e.g. performance requirements) used for the hierarchy

formation will be applied also for the analysis and evaluation of Active Maintenance mechanisms. However, their application will be limited to the locality (neighborhood) of the node that performs the maintenance, as opposed to their global scope during hierarchy formation.

Before any optimization, network hierarchy must generally satisfy certain topological constraints. In creating Dominating Sets, for example, all nodes must always be a bounded number of hops away from a Dominating Node (Clusterhead) (Basagni, 1999). A more general constraint is that every node must be able to reach all other members of its domain via intra-domain links. If constrains are ever violated they must be fixed immediately, else nodes may become unreachable. In some case, constraint violation may result in severe general degradation (e.g., violate the spatial separation in the assignment of frequencies or codes).

3. PASSIVE HIERARCHY MAINTENANCE

Existing approaches to hierarchy maintenance are all passive, in that they are only triggered by events that violate the constraints. Take, for example, the creation of a Dominating Set, with all nodes at most one hop away from a Dominating Node (DN) or Cluster Head (CH). If topology changes causes a node to be more than one hop away for its current DN (i.e., the hierarchy constraints are violated), it either associates with another DN or (if it does not have a DN as a neighbor) a new DN is elected. The exact algorithm depends on the particular protocol. In MACA (Basagni, 1999), for example, there are also constraints about how many DNs can be neighbors and weights used in selecting the best DN. In no case, however, is there any attempt to change the DNs unless a constraint is violated.

In general, it has been shown (Chatterjee, 2002; Manousakis et. al., 2005) that the more relevant to the hierarchy generation objectives is the available information (better quality metrics) during the maintenance phase, the better the "quality" of the hierarchical structure is preserved. For example, if the hierarchy generation objective is the construction of robust domains, then the local maintenance is better to utilize metrics related to the relative speed, direction and position of the participating nodes for the reconstruction of the hierarchy, rather than metrics like node IDs (e.g. lowest ID algorithm). Even if these metrics are available during the passive maintenance, the quality still degrades (Chatterjee, 2002), since passive maintenance focuses mostly on the feasibility of the hierarchy and less on its quality.

4. ACTIVE HIERARCHY MAINTENANCE

This section describes our active local hierarchy maintenance framework, which continuously attempts to preserve the overall quality of hierarchy without imposing too much reconfiguration.

4.1 Active Hierarchy Maintence Overview

Active Local maintenance is performed only from border routers (BRs) desynchronized via the use of *Backoff Timers*. Figure 1 shows the block diagram, which overviews the Active maintenance mechanisms.

At each node, a Neighbor Detection module (see top of Figure 1) generates and maintains a one-hop neighbor list. The Foreign BR Detection module detects if in the list of one-hop neighbors there are nodes of different domains. If a node detects that has BR responsibilities, then checks the status of its "Backoff Timer," responsible for scheduling the Active Maintenance functions on the BR. If the timer has not been initialized yet, then the Backoff Timer Selection module, initializes the timer with a random values generated with respect to a pre-selected probability distribution. On the other hand, if the timer has been initialized before, then the BR checks if the timer has expired.

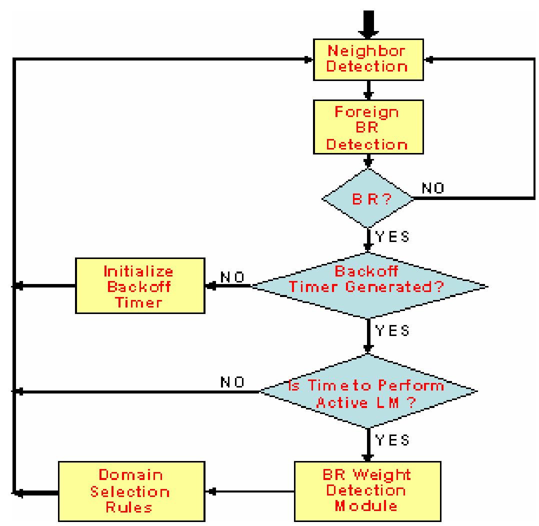

Figure 1. Node Block Diagram of Active Maintenance

If the timer has expired, then the node runs the BR Weight Detection Module. The BR weight depends on the imposed hierarchy generation feasibility constraints, and represents the number of intra-domain nodes that have to migrate if the BR migrates so that the maintained hierarchy is feasible.

The BR then calls the Domain Selection Rules module, which decides what the local reconfiguration will be to improve the hierarchy. A reconfiguration action

involves the migration of BR node and its dependent nodes (e.g. for satisfying the feasibility constraints) to any of its neighboring domains, or even the migration of any of its neighboring BRs along with their dependent nodes.

4.2 Neighbor Detection Module

This module generates and maintains the list of one hop neighbors of the node. The detection of the one hop neighbors is done by either utilizing heartbeat (hello) messages between the nodes or by utilizing available link state information (i.e. from the routing protocol).

4.3 Foreign Border Router (BR) Detection Module

In each domain a node detects if it is able to listen to nodes that belong to different domains. By the exchange of heartbeat messages or residing on link state information, each *BR* generates the list of its neighboring *BRs*, which includes also information about the neighboring domain IDs. This provides the various local domain choices available for preserving or reconstructing the hierarchy formation objectives established by the optimization based hierarchy formation framework.

4.4 Backoff Timer Selection Module

The localized active maintenance has benefits such as low overhead and quick reaction to topology changes. On the other hand, if the local actions synchronize, then they may contribute to the hierarchy quality decline instead of improving it. In particular, as *BRs* are aware only of limited information, multiple of them may decide to perform Active Maintenance concurrently.

To avoid harmful side effects, a desynchronization between the BRs is required. This is a well known problem in ad hoc networks (Chu et. al., 2004). One way to desynchronize is to have *BRs* exchanging coordination information, but such a solution is expensive, slow, and possibly brittle. Our Active Maintenance approach does not need message exchange for BRs desynchronization. A BR, does not immediately perform Active Maintenance. The BRs essentially spread the active maintenance application times in order to avoid colliding decisions. This is achieved with the introduction of *Backoff Timers* similar to that found in other protocols (e.g., a CSMA MAC protocol).

Each *BR* generates a random number $\varphi(\boldsymbol{p})$ with respect to the probability distribution $\boldsymbol{p}$ and initiates an Active Maintenance *Backoff Timer* ξ. By the time $\xi = \varphi(\boldsymbol{p})$, the *BR* performs Active Maintenance. There are three issues that arise due to network dynamics.

1. When $\xi = \varphi(\boldsymbol{p})$ the node may not have anymore *BR* responsibilities. If this happens then the node cancels the scheduled Active Maintenance functions.
2. The *BR* node may change domain during the $\varphi(\boldsymbol{p})$ time units. If the node has retained its *BR* responsibilities, then it performs Active Maintenance when $\xi = \varphi(\boldsymbol{p})$, as scheduled.
3. The propagation of reconfiguration information is not instantaneous. Thus, BR node can decide on a "bad" reconfiguration according to the new topology.

The utilization of the *Backoff Timer* results in distributing over time the Active Maintenance actions taken from different *BRs*. Thus, probabilistically we can minimize or eliminate "colliding" maintenance decisions. In more dynamic networks, we must choose larger *Backoff Timers* to probabilistically force the *BRs* to perform Active Maintenance with up to date information at different points in time.

4.5 Border Router Weight Detection Module

After BR migrations the maintained topology must remain feasible (e.g. satisfy the imposed constraints). For the topological constraints imposed here, the migration of $BR_k^i \in C_i$ from its original domain C_i to a neighboring domain C_j may result to infeasible hierarchy. Specifically, the migration prohibits the intra-domain communications between subset of nodes belonging into BR's original domain with the CH of this domain.

Definition (dependent nodes to BR_k^i): The set of intra-domain paths that connect a node $n_j \in C_i$ to its CH_{C_i} is $P_{n_j}^{C_i} = \left\{ p_{n_j,CH_i}^{C_i} \mid n_j \in C_i, CH_i \in C_i \right\}$ and the set of intra-domain paths of node $n_j \in C_i$ to its CH_{C_i} that include BR_k^i is:

$$P_{n_j,BR_k^i}^{C_i} = \left\{ p_{n_j,CH_i}^{C_i} \mid n_j \in C_i, CH_i \in C_i, BR_k^i \in p_{n_j,CH_i}^{C_i} \right\}$$

If $P_{n_j}^{C_i} = P_{n_j,BR_k^i}^{C_i}$ (e.g. all the intra-domain paths involve BR_k^i) then n_j depends on BR_k^i and is included in $S_{BR_k^i}$, which is the set of nodes $n_j \in C_i$ dependent to BR_k^i, including BR_k^i.

The set $S_{BR_k^i}$ is defined as:

$$S_{BR_k^i} = \left\{ n_j \in C_i \mid P_{n_j}^{C_i} = P_{n_j,BR_k^i}^{C_i} \right\}$$

In order for the migration of BR_k^i to result in feasible hierarchy, every dependent node must migrate with BR_k^i.

The number of migrating nodes defines the weight of BR_k^i.

Definition (weight $w_{BR_k^i}$ of BR_k^i): If the set of nodes $n_j \in C_i$ dependent on BR_k^i is:

$$S_{BR_k^i} = \left\{ n_j \in C_i \mid P_{n_j}^{C_i} = P_{n_j, BR_k^i}^{C_i} \right\}$$

then $w_{BR_k^i} = \left| S_{BR_k^i} \right|$, where $\left| S_{BR_k^i} \right|$ is $S_{BR_k^i}$'s cardinality

Ideally, the dependence information is obtained from a proactive link-state intra-domain routing protocol (Perkins, 2001). However, if the routing protocols is not link state or do not advertise all links (e.g., OLSR), then the information must be gathered by alternative mechanisms.

Apart from the feasibility constraints, there are cases where the weight information is useful for the computation of the localized hierarchical cost – when the hierarchy formation objectives are related to the size of the generated domains. In such cases by computing the weight of a deciding BR, both the feasibility of the reconfigured hierarchy can be preserved and the quality (cost) of the new configuration can be quantified.

4.6 Domain Selection Rules

During the execution of Active Maintenance algorithm, *BRs* can decide if they will stay in their current domain or if they will migrate to a neighboring domain. The decision depends upon the quality (cost) of the resulting local reconfiguration compared to the existing configuration. Local reconfiguration is measured with respect to the objective function f^l, which is inspired by the global objective function f applied during the hierarchy generation phase. Indicative objective functions are provided in Section II. The selection of the most cost effective local reconfiguration during active maintenance has global importance. In combination with the desynchronization of BRs' decisions, better local configurations result into more cost efficient global configurations with respect to the imposed hierarchy formation requirements as they are expressed through the application of the corresponding objective function. Based on Theorem I, the latter holds for the class of indicative objective functions presented in Section II.

Theorem 1 (*From local to global optimality*): Assume the class of objective functions are of the general form:

$$f = \sum_i a_{d_i}^2 \tag{8}$$

where $\alpha \in R$ represents the corresponding metrics and $i \in \mathbf{Z}$ the domains. There is a feasible local domain reconstruction move $\psi_{ij} : d_i \xrightarrow{S_n^{d_i}} d_j$, which involves the migration of a subset of nodes $S_n^{d_i} \in d_i$ from domain d_i to a neighboring domain $d_j \in V^{d_i}$, where:

$$V^{d_i} = \left\{ d_k : d_i \neq d_k \wedge \exists \text{node}_k^{d_k}, \text{node}_i^{d_i} \text{ s.t. } \exists \text{directLink}(\text{node}_k^{d_k}, \text{node}_i^{d_i}) \right\}$$

If ψ_{ij} results on local improvement with respect to (8) then global improvement is achieved.

Proof: There is a feasible move $\psi_{ij} : d_i \xrightarrow{S_n^{d_i}} d_j$, where a node $n_i \in d_i$ and the set of its dependent nodes D^{n_i}:

$$D^{n_i} = \{ n_k : n_k \in d_i, n_k \neq n_i, n_k \text{ gets disconnected from } d_i \text{ if } n_i \text{ migrates to } d_k \neq d_i \}$$

migrate from domain d_i to domain $d_k \in V^{d_i}$. If the domain map (set of all domains) is C, the set $V^{d_i}{}'$ of non-neighboring domains to d_i is:

$$V^{d_i}{}' = C \setminus \left\{ V^{d_i} \bigcup d_i \right\}$$

The cost (8) can be expressed from partial costs as:

$$f = \sum_{d_i} a_{d_i}^2 = a_{d_i}^2 + \sum_{d_z \in V^{d_i}} a_{d_z}^2 + \sum_{d_w \in V^{d_i'}} a_{d_w}^2 = \tag{9}$$

$$= f_{d_i} + f_{V^{d_i}} + f_{V^{d_i'}} = \tag{10}$$

$$= f_{V^{d_i} \bigcup d_i} + f_{V^{d_i'}} \tag{11}$$

where $f_{V^{d_i} \bigcup d_i}$ represents the cost of the local area that involves the domain d_i of the node that initiates the active maintenance and the set of its neighboring domains V^{d_i}. Active maintenance will select the local move $\psi_{ij} : d_i \xrightarrow{S_n^{d_i}} d_j$ (if there is any) that results to better local (partial) reconfiguration with better cost $f^*_{V^{d_i} \bigcup d_i^*}$ such that:

$$f^*_{V^{d_i} \bigcup d_i^*} < f_{V^{d_i} \bigcup d_i} \tag{12}$$

(assuming – without loss of generality – network design requirements are expressed as a minimization problem). The global cost after the move is:

$$f^* = f^*_{V^{d_i} \bigcup d_i^*} + f_{V^{d_i'}} \Rightarrow \tag{13}$$

$$\overset{(5)}{\Rightarrow} f^*_{V^{d_i} \bigcup d_i^*} + f_{V^{d_i'}} < f_{V^{d_i} \bigcup d_i} + f_{V^{d_i'}} = f \Rightarrow f^* < f$$

■QED

Hence, by improving the cost of the local configuration, the global cost improves resulting into better maintained hierarchical structures with respect to the set of pre-specified design objectives.

5. PERFORMANCE EVALUATION

This section evaluates the effectiveness of the proposed Active Maintenance mechanism in the presence of network dynamics (i.e. node mobility). We compare the results with those of Passive Maintenance techniques. Finally, we investigate the effect of *Backoff Timers* selection on the quality of maintained hierarchy.

5.1 Mobility Effect

The performance metric of interest for the any local maintenance framework, is its ability to maintain, or even improve, the quality of hierarchical structure under the existence of topological changes resulting from the mobility of nodes. Here, we assume that the nodes are moving with respect to Random Waypoint Mobility (RWPM) model and their speeds are selected randomly from an interval [*0, maxSpeed*] utilizing the uniform distribution. The *maxSpeed* parameter determines the level of mobility into the network, since the higher it is, the more topological changes are expected to happen.

The quality of the maintained hierarchy is quantified by the objective functions that represent the hierarchy generation requirements. For performance evaluation purposes, we have assumed a representative objective that aims on the formation of balanced size domains (which translates into minimizing routing overhead). The quantification of the objective is represented by (1).

In a bounded area (500m x 500m) network, there are 100 nodes with average node degree of 5. Initially, the nodes are organized in 5 perfectly balanced domains, where each domain has 20 nodes. Then for different values of $\max Speed$ (1m/s, 5m/s and 10m/s), we let the nodes move around in accordance to RWPM model and by applying the Active Local maintenance framework we were recording the fluctuations on the quality (cost) of the maintained hierarchy.

Figure 2 shows the average results of 100 experiments of the effect of mobility on the cost starting from an optimal hierarchy at time 0. Solely Active Local Maintenance is applied to preserve the balanced size domains. As expected, the lower the rate of topology changes the slower the hierarchy is degraded (cost increases). More interesting observations can be drawn, however, when we let the Active Maintenance applied for long periods of time. We find that, after some time, the quality of the actively maintained hierarchy is not affected from the degree of mobility. Furthermore the quality of the maintained hierarchy is acceptable (cost increase is less than 50%). This is in contrast to Passive Maintenance, which can quickly result in very poor hierarchy quality (cost increase up to 400% as it is presented in the following subsection).

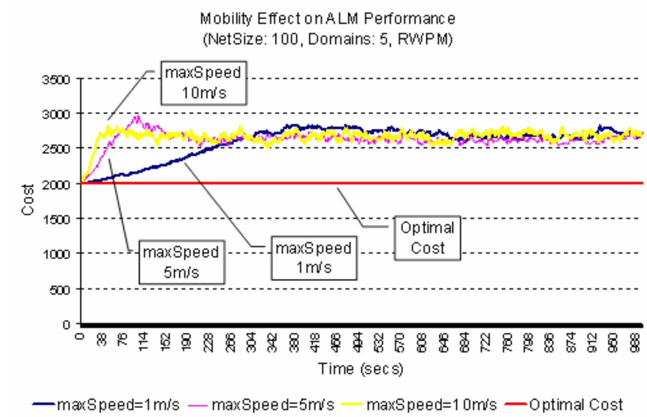

Figure 2. Mobility effect on Active Maintenance

5.2 Active vs. Passive Distributed Maintenance

This subsection compares the passive methods (Manousakis et. al., 2005) with the Active Local maintenance framework, with respect to hierarchy quality (cost) preservation in dynamic (mobile) networks. The experimental evaluation parameters are the same as those above, except that the RWPM had *maxSpeed* of 10m/s. The Backoff Timer follows a uniform distribution $U \sim [0, \vartheta]$ with $\vartheta = 1s$. Figure 3 shows the average outcome of 100 experiments.

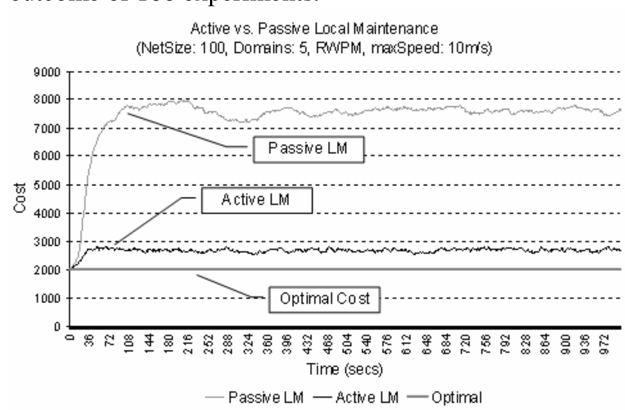

Figure 3. Active vs. Passive Maintenance with respect to hierarchy quality preservation when RWPM is applied.

The Active Local maintenance maintains the high quality (cost increases less than 50%) of the hierarchy with respect to the generation objectives (balanced size domains) as opposed to Passive Local Maintenance, which results in rapid degradation of the hierarchy quality (cost increases by almost 400%) into levels where the formation objectives are far from satisfied. Thus, in the case of Passive Local maintenance the expensive

hierarchy generation algorithm has to be frequently reapplied for reconstructing the hierarchy quality. That will result in high overhead or even in inability of the hierarchy formation algorithm to capture the dynamics of the network, in scenarios of high mobility (Manousakis et. al., 2004).

5.3 Active Maintenance Decisions Desynchronization

For the BRs desynchronization, we have utilized the Backoff Timers approach. Instead of passively waiting for events (i.e., feasibility constraint violations), or for a periodic timer, a BR that performs Active maintenance, schedules its maintenance decisions in the future based on a pre-specified distribution. This approach requires no synchronization with other BRs, thus has zero overhead with respect to coordination communication messages. On the other hand the trade off is that when a BR delays the application of Active maintenance, it might miss the opportunity to improve immediately the hierarchical structure.

Table III. Experimental Scenarios

Scenario	Mobility Model	Max Speed (m/s)	Distribution	Mean Value
S1	RWPM	1	Exponential	1
S2	RWPM	1	Exponential	10
S3	RWPM	1	Uniform	1
S4	RWPM	1	Uniform	10
S5	RWPM	10	Exponential	1
S6	RWPM	10	Exponential	10
S7	RWPM	10	Uniform	1
S8	RWPM	10	Uniform	10

Intuitively, an important parameter responsible for the stability and goodness of the Active maintenance is the selection and distribution of *Backoff Timers*. In order to understand the effect of Backoff times we investigated two different distributions: Uniform and the Exponential. Within each of these distributions we investigated different parameters (e.g., mean value). For each approach (listed in Table III), we studied the effect that they have on preserving the quality of hierarchical structure.

Figures 4 and 5 provide simulation results, which have been averaged out of 100 experiments. In each scenario the hierarchy formation objective is the configuration of 5 balanced size domains (1) of 20 nodes each. At time t=0s the hierarchy is formed by applying the optimization-based framework (global optimization). Then the proposed Active maintenance is applied for 1000 consecutive network seconds. The figures represent the preservation of the hierarchy quality (e.g. the lower the value, the better the configuration) with respect to the pre-specified objectives. The hierarchy quality is quantified using (1).

The main observation from the results is that the distribution applied does not affect significantly the effect of the maintenance mechanism. The mean value selected for the distribution at hand affects – not significantly though – the reaction time of the maintenance with respect to mobility levels. Specifically, the lower the mean value, the faster the reaction time towards improving the hierarchy quality, even though the probability of colliding Active maintenance decisions from different BRs gets higher.

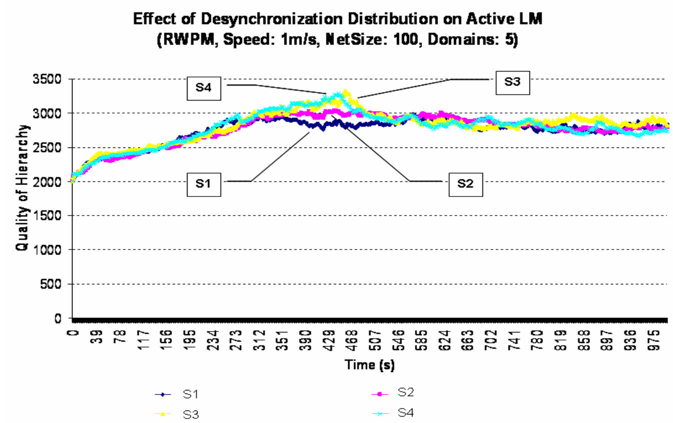

Figure 4. Effect of Backoff Timers selection on the maintenance of hierarchy quality in the presence of slow network dynamics (max node speed 1m/s)

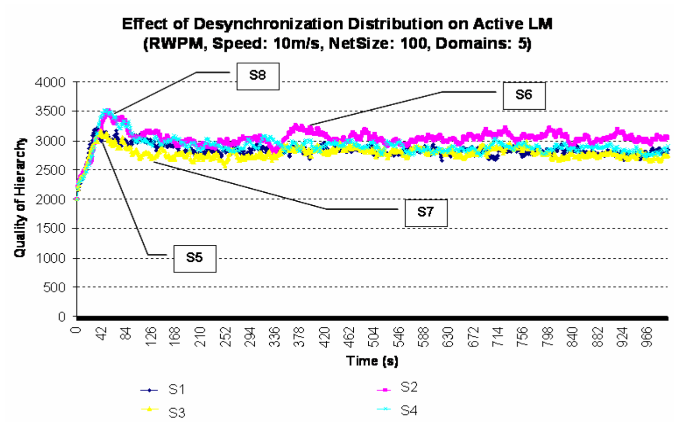

Figure 5. Effect of Backoff Timers selection on the maintenance of hierarchy quality in the presence of faster network dynamics (max node speed 10m/s).

5.4 Distributed Reoptimization of Hierarchy Quality

Passive local maintenance generally moves asymptotically towards some feasible but highly non-optimal solution. If we assume that due to Passive maintenance, the hierarchy has become non-optimal (e.g., has cost close to 2500 in Figure 6), then only the application of the global optimization framework can

restore the hierarchy quality.

As opposed to Passive Local maintenance, the application of Active Local maintenance, due to its active nature and focus on hierarchy quality, is able of improving the quality of hierarchical structure with respect to the pre-specified set of hierarchy formation objectives. Passive Local maintenance does not have this ability, since it is event-driven focusing on the hierarchy feasibility (a node has to become infeasible to trigger the maintenance, otherwise no action is taken and no quality improvement is possible).

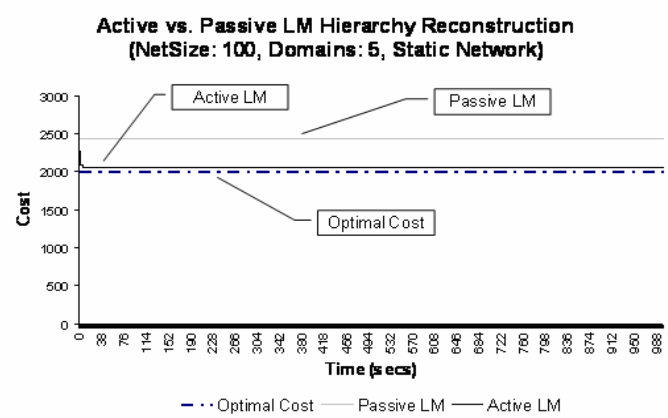

Figure 6. Ability of Active Maintenance to distributively reoptimize the quality of hierarchy as opposed to Passive Maintenance

Figure 6 is representative of the hierarchy quality reconstruction ability of Active Local maintenance framework. It represents the evolution of hierarchy quality when a static network of 100 nodes is initially organized in 5 unbalanced feasible domains and then Passive Local maintenance and Active Local maintenance are applied. The Active Local Maintenance reacts rapidly to the low quality hierarchy. With successive active local maintenance decisions from BRs, the hierarchy quality is reconstructed very close to the optimal cost (close to 2000 with respect to function (1)). On the other hand Passive Local maintenance cannot improve the unbalanced hierarchy, since the static network (organized in feasible hierarchical structure) does not generate hierarchy infeasibility, which will trigger the Passive Local Maintenance to react.

CONCLUSIONS

We present a new class of Local Maintenance algorithms that, unlike the Passive Local Maintenance algorithms, are not triggered only by events that violate the existing hierarchy (e.g. infeasibility). The new Active Local Maintenance algorithms actively and continuously maintains the hierarchy quality. The results show that Active Local Maintenance actually manages to preserve good hierarchy quality (cost increase is less than 50% of the global optimal), independently of the network dynamics. To the contrary, Passive Local maintenance shows rapid quality degradation (cost increase up to 400%).

Interestingly, the selection of Backoff Timer distributions has little impact on performance. Further work is needed to understand why the selection of Backoff Timer approach chosen for our experiments had so little impact on performance.

REFERENCES

Basagni S., 1999: Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks, Proceedings of Vehicular Technology Conference, VTC 1999-Fall, Vol. 2, pp. 889-893

Bao L. and Garcia-Luna-Aceves J.J., 2003: Topology Management in Ad Hoc Networks," The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), Annapolis, Maryland, June 2003.

Chatterjee M., Das S. K., Turgut D., "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks, " Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, April 2002, pp. 193-204

Chu W., Colbourn C. J., Syrotiuk V. R., 2004: Topology Transparent Scheduling, Synchronization, and Maximum Delay, 18th International Parallel and Distributed Processing Symposium (IPDPS'04), 2004.

Krishnan R., Ramanathan R. and Steenstrup M., 1999: Optimization Algorithms for Large Self-Structuring Networks, Proceeding of IEEE INFOCOM '99, New York, USA, March 1999.

Lin R. and Gerla M., 1997: Adaptive Clustering for Mobile Wireless Networks, IEEE Journal on Selected Areas in Communications, pp. 1265-1275.

Manousakis K., Baras J.S., 2004: Dynamic Clustering of Self Configured Adhoc Networks Based on Mobility, Conference on Information Sciences and Systems (CISS), Princeton, NJ, March 2004.

Manousakis K., McAuley A. J., Morera R., 2004: Applying Simulated Annealing for Domain Generation in Ad Hoc Networks, *IEEE International Conference on Communications (ICC) 2004*, Paris, France, June 2004.

Manousakis K., McAuley A. J., Morera R., Baras J. S., 2005: Design Framework for Hierarchy Maintenance Algorithms in Mobile Ad hoc Networks, *IEEE MILCOM 2005*, Atlantic City, NJ, October 2005.

Perkins C.E., 2001: Ad Hoc Networking, Addison-Wesley, Boston, MA, USA, January, 2001

Ramalingam A., Subramani S., and Perumalsamy K., 2002: Associativity based cluster formation and cluster management in ad hoc networks, in Proc of HIPC 2002, Bangalore, India.
